



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/535,573	03/27/2000	Robert A. Foster	M-4540-1C us	3655
7590	03/02/2005		EXAMINER	
MACPHERSON KWOK CHEN & HEID LLP 1726 TECHNOLOGY DRIVE SUITE 226 SAN JOSE, CA 95110			NGUYEN, CUONG H	
			ART UNIT	PAPER NUMBER
			3661	

DATE MAILED: 03/02/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

<i>Office Action Summary</i>	Application No.	Applicant(s)
	09/535,573	FOSTER, ROBERT A.
Examiner	Art Unit	
CUONG H. NGUYEN	3661	

*– The MAILING DATE of this communication appears on the cover sheet with the correspondence address –*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1) Responsive to communication(s) filed on 18 October 2004.  
 2a) This action is FINAL.                            2b) This action is non-final.  
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4) Claim(s) 1-86 is/are pending in the application.  
 4a) Of the above claim(s) 47-86 is/are withdrawn from consideration.  
 5) Claim(s) \_\_\_\_\_ is/are allowed.  
 6) Claim(s) 47-86 is/are rejected.  
 7) Claim(s) \_\_\_\_\_ is/are objected to.  
 8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

9) The specification is objected to by the Examiner.  
 10) The drawing(s) filed on \_\_\_\_\_ is/are: a) accepted or b) objected to by the Examiner.  
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).  
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
 a) All    b) Some \* c) None of:  
 1. Certified copies of the priority documents have been received.  
 2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) Notice of References Cited (PTO-892)  
 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)  
 3) Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
 Paper No(s)/Mail Date \_\_\_\_\_

4) Interview Summary (PTO-413)  
 Paper No(s)/Mail Date \_\_\_\_\_  
 5) Notice of Informal Patent Application (PTO-152)  
 6) Other: \_\_\_\_\_

**DETAILED ACTION**

1. This Office Action is the answer to the amendment received on 10/18/2004.
2. Claims 47-86 are pending in this application.

**Response**

3. Applicant's amendment and arguments have been fully considered but they are not persuasive because the claimed language read-on cited references.
4. In page 19 or the paper (10/22/04) applicant argues "The Examiner is well aware that Moore, Burt, Rothstein and Claus at best suggest general notions of financial transaction, and not at all the specific data structures recited in Applicant's claims"; the examiner respectfully disagrees, the examiner submits independent claim 47 "broadly" suggests steps of providing a database, comprising:

- (a) creating a transaction instance;
- (b) creating a production service instance;
- © creating a billing service instance;
- (d) pricing said transaction "based on" above instances (there are "inherent" predetermined links among a transaction instance, a production service instance, and a billing service instance – pricing a transaction "based on" related components is not inventive).

The combination of cited Moore et al. and Burt et al. suggest above steps (newly added step (d) is obvious with Moore et al. 6:1-4).

- Since there is no explanation in the claim of what an instance besides naming different instances, this concept was read-on by cited references (the examiner respectfully requests an amended claim to show how to create, and how to price a transaction other than claiming "broad" steps of creating instances); applicant argues on page 19 that "The combination of general notions of financial transactions and specific programming techniques simply does not disclose or suggest specific data structures necessary for pricing complex transaction"; the examiner respectfully submits that he does not see any distinguished meanings about above "specific data structure" in claim 47 except calling different names/instances (please note also that claim 47 is a method claim - not a system that requires "specific data structure"). The applicant requests to reinstate the Appeal Brief; however, in the Appeal Brief, the current rejection rationales MUST be argued, not the previously rejected rationales. The applicant confirms that "instance" in this invention has a broad, normal meanings (see "REMARKS" submitted on 10/22/2004); therefore, the cited references already comprise that "broad, and normal" meanings because this is a specific application of the term "instance" with its actual meanings. On page 19, the applicant confirms "None of Applicant's claims is limited by any structure of any programming language, object-oriented or otherwise"; this admission clearly

indicates that pending claims do not conform to 35 USC 101 "technological art" requirements.

In the previous Office Action, the examiner states "claim 47 may not be concrete and tangible when a computer is not running" (i.e., claimed steps do not exist if no execution), he means that the pending claimed invention MUST require an operational computer for performing those steps. The 35 USC 101 panel advised on Wed., 1/12/2005 that the amended claim 47 is still not statutory; a requirement for technological art on 35 USC 101 (previous rejections were not included because "technological art" not enforced).

In the response received on 10/22/2004, the applicant confirmed that the claimed term "instance" has no special meanings besides its original dictionary definition. Therefore, a broader interpretation of claim 47 is established (previous interpretation - Office Action issued on 6/18/2004 - requires a "narrower" application of computer OOP for this special reserved word "instance").

5. The term of "being linked" is taught by **Burt** et al. as using "connection layer" and "connection instance" (see **Burt** et al., Fig.5),. The examiner submits that **Burt** et al. teach a relational model as claimed (see **Burt** et al., Fig.5); "relational model is a data model in which the data is organized

in relations (tables). This is the model implemented in most modern database management system"; therefore, banking transactions and related pricing were known to implement this model for the pending claimed relational structures (such use of a relational database management software have been applied for banking transactions, see **Moore et al.**, the abstract, Figs.3,4, 10:5-34, 23:27-61 e.g., these para. indicate relationships of an object and access type with the value in the object instance entity).65. The examiner also submits that suggestions for "Categorization of purchased items for each transaction by a smart card" had been discussed by **Claus et al.** in their patent (see **Claus**, claim 1, and 2:15 to 3:22), and a relational database of items/instances for different transactions were done in a financial software program.

6. The Microsoft Computer Dictionary (published in 1996) defines a standardized meaning of a database wherein data components are linked together within that database as followings: linked list: In programming, a list of elements of a data structure connected by pointers. A singly linked list has one pointer in doubly linked list has two pointers in each node pointing to the next and previous nodes. In a circular list, the first and last nodes of the list are linked together; and link: To produce an executable program from compiled modules

(programs, routines, or libraries) by merging the object code (assembly language object code, executable machine code) of the program and resolving interconnecting references (such as a library routine called by a program), or to connect two elements in a data structure by using index variables (index: A listing of keywords and associated data that point to the location of more comprehensive information, such as files and records on a disk/record keys in a database), or pointer variables (pointer: In programming and information processing, a variable that contains the memory location (address) of some data rather than data itself). The act of linking data/items from different parts in a database is in cited references of **Moore et al.**, **Burt et al.**, **Rothstein**, **Clause et al.**, and they are a fundamental knowledge in database structure of OOPs; from that available computer programming knowledge the applicant uses it to apply for a specific use (i.e. for pricing transactions). Therefore, the invention does not teach any new inventive concept according to cited references.

7. The examiner respectfully submits that claims 47, and 68 comprise elements of a relational database structure, would utilize an instance variable in its object-oriented program (i.e., an **instance** is an instantiated object of a particular

class), an *object* is something that can have properties and relations).

8. At the end, on pg.5, 1st para., the applicant argues that cited references do not teach a "client instance" and a "market segment instance".

The examiner submits that one with ordinary skill in the art would understand that in an object-oriented program (e.g., Java, Visual Basic, C++ .etc.):

- an (entity) instance could be a client instance; an entity instance could be a market segment instance because in OOP, "instance" is a variable: an instance is an *instantiated object* of a particular class.

The examiner submits that cited prior art's limitations are not necessary spelled-out exactly claimed languages; analogous interpretations based on definition for functions of those terms show that such claimed languages would be obvious for meaningful modifications in OOP using in cited art's situations.

9. All claimed limitations have been known since events for pricing transactions always "link" to related objects in a relational database. As the examiner presents that the claimed subject matter is obvious with one of skills in the art, different "instances" in above claims may be defined according to the use of a particular "instance" in an object-oriented

program, in relation to the "class" to which it belongs; in other words, instance variable is just a variable associated with an event/action/instance of a class in OOPs (a class is a template for a group of objects an object such as: client, market segment with similar behaviour, and a common inheritance).

The main subject matter of this invention is about "pricing transaction"; besides cited references, Sprague et al. (US Pat. 5,247,575) also teach that in col. 19, line 63 to col. 20 line 18, col. 22 lines 59-66, and col.23, lines 46-65; Hartrick et al. (US Pat. 5,532,920) also suggests that in Figs. 10-11.

- Peterson (U.S. Pat. 6,324,522) also discloses the claimed invention - for a best price fee - including receiving a request for a real-time price quote for a transaction (purchasing an item); the request occurring within a billing cycle (billing cycles are inherent in commercial enterprises); determining a total (the sum of all goods, products, or services purchased); determining a billing service (postal mail, internet, etc. inherent in commercial operations, the bill has to get to the customer somehow); the first attribute being price per unit with available volume discounts); apportioning the price to the transaction (the seller has to pay his supplier based upon the number of units he in turn received).

***Claim Rejections - 35 USC §101***

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requires of this title.

10. Since USPTO are examining applications for utility patents, the claims must be directed to systems, methods or articles of manufacture that have a clear utility. See MPEP 706.03(a) for example. Over the years, numerous court decisions have analyzed the content of various claim language for meaningful, useful differences in structure or acts performed between the claims and the prior art.

11. **Claims 47-67 are rejected under 35 U.S.C. 101 because, the claimed invention is directed to non-statutory subject matter.**

They contain computer-per-se materials according to the preamble (i.e., "In a computer-readable medium,"). The claim subject matter MUST be useful, in contrast, claim 47 is ONLY useful when a computer is not running a claimed method (in claim 47) can not be derived (i.e., merely claiming a floppy disk with claimed instructions) - *In State Street Bank & Trust Co. v. Signature Financial Group, Inc.*, 47 USPQ2D 1596, 1601-02 (Fed. Cir. 1998), the Court determined that when an abstract idea is reduced to a practical application, the abstract idea no longer stands alone if it produces a "useful, concrete and tangible

result" - because claim 47 may not be concrete and tangible when a computer is not running to perform claimed steps (without a running computer, the claimed steps are not existed), it is non-statutory.

The invention MUST be a concrete idea; however, recited in those pending claims' limitations are not concrete material when a computer is not executing instructions in that computer-readable medium (i.e., there is NO physical structural relationships of said system's computer's components, there is NO link, no creation of different OO instances).

These claims contain abstract ideas; (i.e., containing computer-per-se materials, although a system for pricing transactions is claimed). The claimed "virtual" system has all "virtual" structural components wherein those components are computer instructions, i.e., a means for creating a transaction instance, means for creating a first production service instance, means for creating a billing service instance .etc.; therefore, it is an abstract idea to one of ordinary skill in the art to recreate the claimed system for pricing transactions.

The invention as recited in these claims is merely an abstract idea that is not within the technological arts. Mere abstract ideas that do not apply, involve, use the technological arts fail to promote the "progress of science and the useful

arts" (i.e., the physical sciences as opposed to social sciences, for example) and therefore are found to be non-statutory subject matter [see Bowman (BPAI), 61 USPQ2d 1669, 6/12/2001].

Even mere recitation in the preamble or mere suggestion in the claim that a machine is performing some or all of the steps in the method is NOT ENOUGH to place claimed invention in the technological arts. The body of the claim must unambiguously recite that a machine/apparatus is performing the step(s) and/or is integrally involved in the process (i.e., a computer-implemented method) for the achieved effect (i.e., level of involvement, use, or advancement).

The phrase "technological arts" is synonymous with the phrase "useful arts" as it appears in Article I, Section 8 of the Constitution, *In re Waldbaum*, 173 USPQ 430 (CCPA 1972). For a claim to be statutory, it must be in the technological arts. *In re Musgrave*, 167 USPQ 280 (CCPA 1970) and *In re Johnston*, 183 USPQ 172 (CCPA 1974).

12. Claims 47-67 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

As an initial matter, the United States Constitution under Art. I, §8, cl. 8 gave Congress the power to "[p]romote the progress of science and useful arts, by securing for limited times to authors and inventors the exclusive right to their respective

writings and discoveries". In carrying out this power, Congress authorized under 35 U.S.C. §101 a grant of a patent to "[w]hoever invents or discovers any new and useful process, machine, manufacture, or composition or matter, or any new and useful improvement thereof." Therefore, a fundamental premise is that a patent is a statutorily created vehicle for Congress to confer an exclusive right to the inventors for "inventions" that promote the progress of "science and the useful arts". The phrase "technological arts" has been created and used by the courts to offer another view of the term "useful arts". See *In re Musgrave*, 167 USPQ (BNA) 280 (CCPA 1970). Hence, the first test of whether an invention is eligible for a patent is to determine if the invention is within the "technological arts".

Further, despite the express language of §101, several judicially created exceptions have been established to exclude certain subject matter as being patentable subject matter covered by §101. These exceptions include "laws of nature", "natural phenomena", and "abstract ideas". See *Diamond v. Diehr*, 450, U.S. 175, 185, 209 USPQ (BNA) 1, 7 (1981). However, courts have found that even if an invention incorporates abstract ideas, such as mathematical algorithms, the invention may nevertheless be statutory subject matter if the invention as a whole produces a "useful, concrete and tangible result." See

*State Street Bank & Trust Co. v. Signature Financial Group, Inc.* 149 F.3d 1368, 1973, 47 USPQ2d (BNA) 1596 (Fed. Cir. 1998).

This "two prong" test was evident when the Court of Customs and Patent Appeals (CCPA) decided an appeal from the Board of Patent Appeals and Interferences (BPAI). See *In re Toma*, 197 USPQ (BNA) 852 (CCPA 1978). In *Toma*, the court held that the recited mathematical algorithm did not render the claim as a whole nonstatutory using the Freeman Walter-Abele test as applied to *Gottschalk v. Benson*, 409 U.S. 63, 175 USPQ (BNA) 673 (1972). Additionally, the court decided separately on the issue of the "technological arts". The court developed a "technological arts" analysis:

The "technological" or "useful" arts inquiry must focus on whether the claimed subject matter ... is statutory, not on whether the product of the claimed subject matter... is statutory, not on whether the prior art which the claimed subject matter purports to replace ... is statutory, and not on whether the claimed subject matter is presently perceived to be an improvement over the prior art, e.g., whether it "enhances" the operation of a machine. *In re Toma* at 857.

In *Toma*, the claimed invention was a computer program for translating a source human language (e.g., Russian) into a target human language (e.g., English). The court found that the

claimed computer implemented process was within the "technological art" because the claimed invention was an operation being performed by a computer within a computer.

The decision in *State Street Bank & Trust Co. v. Signature Financial Group, Inc.* never addressed this prong of the test. In *State Street Bank & Trust Co.*, the court found that the "mathematical exception" using the Freeman-Walter-Abele test has little, if any, application to determining the presence of statutory subject matter but rather, statutory subject matter should be based on whether the operation produces a "useful, concrete and tangible result". See *State Street Bank & Trust Co.* at 1374. Furthermore, the court found that there was no "business method exception" since the court decisions that purported to create such exceptions were based on novelty or lack of enablement issues and not on statutory grounds. Therefore, the court held that "[w]hether the patent's claims are too broad to be patentable is not to be judged under §101, but rather under §§102, 103 and 112." See *State Street Bank & Trust Co.* at 1377. Both of these analysis goes towards whether the claimed invention is non-statutory because of the presence of an abstract idea. Indeed, *State Street* abolished the Freeman-Walter-Abele test used in *Toma*. However, *State Street* never addressed the second part of the analysis, i.e., the "technological arts" test established in

Toma because the invention in State Street (i.e., a computerized system for determining the year-end income, expense, and capital gain or loss for the portfolio) was already determined to be within the technological arts under the Toma test. This dichotomy has been recently acknowledged by the Board of Patent Appeals and Interferences (BPAI) in affirming a §101 rejection finding the claimed invention to be non-statutory. See *Ex parte Bowman*, 61 USPQ2d (BNA) 1669 (BdPatApp&Int 2001).

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. §103(a), which forms the basis for all obviousness rejections set forth in this Office Action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negative by the manner in which the invention was made.

13. **Claim 47 is rejected under 35 U.S.C. §103(a) are rejected under 35 U.S.C. §103(a) as being unpatentable over Moore et al. (US Pat. 5,630,127), in view of Burt et al. (US Pat. 5,682,482).**

It is directed to steps of:

- (a) creating a “transaction” instance;
- (b) creating a “production service” instance;
- © creating a “billing service” instance;

(d) pricing said transaction "based on" above instances (there are "inherent" predetermined links among a transaction instance, a production service instance, and a billing service instance – pricing a transaction "based on" related components is not inventive).

The combination of cited Moore et al. and Burt et al. suggest above steps (newly added step (d) is obvious with Moore et al. 6:1-4).

Here each instance having data identifying particular items such as transaction, production service, billing service. These data qualifies as non-functional descriptive material.

These claimed descriptive material are not functionally related to a substrate (e.g., a computer-readable medium). Rather those are just "being held" in the medium. As a result, those data can be called non-functional descriptive material and does not limit the claim.

Moore et al. teach that a rule-based application structure could be a relational database where records of a transaction are related/linked to each other (see Moore, the abstract, and Figs. 3,4). Moore et al. teach that: service instances linking to transaction instances; and creating a billing service instance linked to a service instance with relation instance (see Moore, "FIG. 4 is an object instance table." 6:54-59 "An example of this table is shown in FIG. 3. The names or "objects" are shown in the columns "OBJECT" 302, "OBJECT1" 304 and "OBJECT2" 308. These names or "objects" stand for a multitude of particular

instances of the data, any of which can be retrieved by specifying the identifiers of the entities listed above which would focus the access on a particular representation value."; and Moore et al., 10:5-19, and 10:45-55 "An additional feature of the GRMS architecture is the placement of the GRMS processor on the Business Professional's workstation 118 along with the Object Table 300, and the programs defined in the object table 300. Since the object instance table 400 is also present, the Business Professional can change values in the Object Instance table (via GRMS screens and functions) and reprocess the report on the workstation. All object accesses will be satisfied by the Object Instance table function and therefore, the CMIM database 224 is not needed for this "What if" analysis reporting."; in OOP, "instance" is a variable name e.g., service instance, relation instance .etc.).

Moore et al. teach about a financial institution, and a single transaction can generate many object instances (see Moore et al., 1:21-30, and 28:60-62 "A single GRMS transaction can generate many object instances"); Moore et al. do not explicitly disclose that financial transaction functions are connected together.

Burt et al. further disclose a system with related functions including financial transaction functions connecting together (e.g. see Burt et al., Fig. 5, the abstract, 4:25-27, and 25:2-16), comprising:

- creating a transaction instance corresponding to a financial transaction (e.g. see Burt et al., Fig.5, the abstract, col.6 lines 1-14, and col.21 lines 42-59).

The examiner respectfully submits that because Moore et al. teach applications using OOP macros wherein "instance" is a variable instance - an instance is a single occurrence of a class -, it would be obvious for the analogous use of macros: "transaction instance", "service instance", and "billing service instance". Artisan would recognize that a total price for a transaction including any changes of those instance variables (i.e., pricing a transaction "based on": a transaction instance, production service instance, and billing service instance).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to create different instances as shown in Moore et al., and Burt et al. because a total price for a transaction would take into account all predetermined pricing components related to said transaction (see *in re Gulack*, 703 F.2d 1381, 1385, 217 USPQ 401, 404 (Fed. Cir. 1983)).

**14. Claim 68 is rejected under 35 U.S.C. §103(a) as being unpatentable over Moore et al. (US Pat. 5,630,127), in view of Burt et al. (US Pat. 5,682,482).**

It is directed to a data processing system that comprises a means for creating a transaction event, a means for creating a production/service event, the linking between said events, and a means for compute a price using said events; this claim has

similar limitations as of claim 47 except steps of claim 47 are now written in a means-plus-function claim. Thus it is also rejected for the same rationales and references of Burt et al., and More et al., as above rejected independent claim 47.

**15. Claims 75-77, 56-57 are rejected under 35 U.S.C. §103(a) are rejected under 35 U.S.C. §103(a) as being unpatentable over Moore et al. (US Pat. 5,630,127), in view of Burt et al. (US Pat. 5,682,482).**

A. Re. to claim 75: The rationales and references for rejecting claim 68 are incorporated.

Moore et al. teach that an OOP software is used for creating different instances (i.e., a fourth relation instance) that links different instances (e.g., linking a transaction instance to an entity instance), (see Moore et al. Fig.4, and col.10 lines 25-55).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to combine specific applications of Moore et al., Burt et al., in financial transaction (for different applications using relational database) because they all suggest a systematic method that use "instance" to track all of the components of costs and fees each time a financial transaction is processed. It has been recognized that a finance system would be able to measure

profitability in a flexible manner and to measure the impact of any changes from banking clients by tracking those variables.

B. Ref. to claims 76, 77, 56, and 57:

The rationales and references for rejecting claim 68 are incorporated.

Burt et al. further teach about storing/retrieving relation instances in relation instance table (e.g., see Burt et al., claim 5 - this claim indicates that different rules for objects/instances are stored in tables, and can be retrieved from those tables); and creating a second entity instance related to first entity instance (e.g. see Burt et al., Fig. 4 - this figure indicates that different instances have relationships).

**16. Claims 79, 59, 83, 63, 65, and 84 are rejected under 35 U.S.C. §103(a) are rejected under 35 U.S.C. §103(a) as being unpatentable over Moore et al. (US Pat. 5,630,127), in view of Burt et al. (US Pat. 5,682,482).**

The rationales for rejection of claims 68 are incorporated herein.

Moore et al. and Burt et al. also teach :

- an OOP for creating an entity instance relating to another instance (e.g., a transaction instance - see Moore et al., Fig.4, 6:54-59, 10:5-19, 45-55);
- an OOP for creating an entity instance relating to above entity instance;

Moore et al. also teach a means for creating a price table instance related to an entity instance (see Moore et al., Fig.4); the examiner submits that in OOP variable instances can be created, and can be defined to relate to each other. (The claimed phrase of "wherein a price table instance contains a price for a billing service instance" is a specific but fundamental application of instance variables in OOP).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to combine specific applications of Moore et al., and Burt et al., for financial transaction (using relational database) because they all suggest a systematic method that use "instance" to track/pricing components of costs and fees each time a financial transaction is processed. Artisan would recognize that a finance system would be a flexible application to measure the impact of any changes from financial transactions by tracking those instance variables (i.e., pricing a transaction based on a transaction instance, production service instance, and billing service instance).

17. Claims 60, and 80 are rejected under 35 U.S.C. §103(a) are rejected under 35 U.S.C. §103(a) as being unpatentable over Moore et al. (US Pat. 5,630,127), in view of Burt et al. (US Pat. 5,682,482).

The examiner's position is that in an OOP software, it is obvious to define that "price table instance is a cost table instance, and a price is a cost" since it is merely defined as a variable instance in Moore et al.'s transaction.

The rationales and references for rejecting claim 79 are incorporated.

The examiner submits that a price table instance could be defined as a cost table instance, and said price could be a cost; or a price table instance could be defined as a fee table instance since price/fee table instance is just a sample instance data structure.

**18. Claims 81, and 61 are rejected under 35 U.S.C. §103(a) are rejected under 35 U.S.C. §103(a) as being unpatentable over Moore et al. (US Pat. 5,630,127), in view of Burt et al. (US Pat. 5,682,482).**

The rationales and references for rejecting claim 79 are incorporated.

- The examiner submits that one of ordinary skill in the art would recognize that a price is understood as a fee (e.g., a parking fee in a parking garage is similar as a parking price/cost).

**19. Re. To Claims 82, and 62: They are rejected under 35 U.S.C. §103(a) are rejected under 35 U.S.C. §103(a) as being unpatentable over Moore et al. (US Pat. 5,630,127), in view of Burt et al. (US Pat. 5,682,482).**

The rationales and references for rejecting claim 81 are incorporated.

Moore et al., and Burt et al. use OOP to teach relations/linking between variable instances. Because their instances are variable, it is obvious to name them meaningfully to each specific application.

The uses of a relational database in cited prior art teach a step of creating a cost table instance related to a fee table instance by a relation instance.

Moore et al., and Burt et al. do not specifically disclose that "a cost table instance related to a fee table instance by a relation instance", the examiner respectfully submits that any OOP application having a characteristic of linking/relating between object instances (e.g., see the admission from contents of claims 66, 85 wherein the applicant confirms that "entity instance" is a variable).

**20. Claims 64, 66, and 85 are rejected under 35 U.S.C. §103(a) are rejected under 35 U.S.C. §103(a) as being unpatentable over Moore et al. (US Pat. 5,630,127), in view of Burt et al. (US Pat. 5,682,482).**

The rationales and references for rejecting claim 84 are incorporated.

The examiner submits that because an instance is defined as a variable in OOP, an entity instance can be defined as an

account instance, a client instance, or can be defined as a market segment instance (see Burt, the abstract, claims 1-2).

By contents of the pending claims 66, 85, the applicant admits that an entity instance is a variable instance, since an entity instant can be used as an account instant or as a client instant.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to combine specific applications of Moore et al., and Burt et al., in OOP financial transaction because they all suggest a systematic method that use "instance" in an OOP to track components of costs and fees each time a financial transaction is processed. Artisan would recognize that an instant in OOP would be a variable to measure the impact of any changes from financial transactions by tracking those instance variables.

**21. Claims 48, 50-53, 58, 69, 71-75, and 83 are rejected under 35 U.S.C. §103(a) are rejected under 35 U.S.C. §103(a) as being unpatentable over Moore et al. (US Pat. 5,630,127), in view of Burt et al. (US Pat. 5,682,482).**

The rationales and references for rejecting claim 68 are incorporated.

Moore et al. obviously suggest a step of storing a transaction instance/an account instance/a client instance, a

production service instance, a settlement service instance, and a billing service instance in an entity instance table, and they are inherently "link"/"relate" together as a functional data structure (e.g. see Moore et al. Fig.4, and col.10 lines 25-55).

A. Re. To claim 48: This claim is directed to a method of pricing transactions containing similar limitations as in "system" claim 69. Therefore, similar rationales and references set forth are also used for a 35 USC 103(a) rejection.

B. Re. To claims 69, 71-74: Theses claims are directed to a system of creating a "billing service instance" that link to a transaction instance by a relation instance. The examiner submits that this is an available function of an OOP used by Moore et al. Therefore, similar rationales and references set forth are also used for a 35 USC 103(a) rejection.

C. Re. To claim 50: This claim is directed to a method of pricing transactions containing similar limitations as in "system" claim 71; therefore, similar rationales and references set forth are also used for a 35 USC 103(a) rejection.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to combine specific applications of Moore et al., and Burt et al., in OOP financial transaction because they all suggest a systematic method that use "instance" in an OOP to track components of costs and fees each time a financial transaction is processed. Artisan would

recognize that an instant in OOP would be a variable to measure the impact of any changes from financial transactions by tracking those instance variables.

22. **Claim 78 is rejected under 35 U.S.C. §103(a) are rejected under 35 U.S.C. §103(a) as being unpatentable over Moore et al. (US Pat. 5,630,127), in view of Burt et al. (US Pat. 5,682,482).**

The rationales and references for rejection of claim 68 are incorporated.

- means for creating a settlement service instance linked to said billing service instance by a third relation instance. Moore et al. teach that an OOP software is used for created different instances (i.e., a settlement service instance) that link/relate (using a relation instance) with another instance (i.e., a billing service instance), (see Moore et al. Fig.4, and col.10 lines 25-55).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to combine specific applications to combine Moore et al., Burt et al., in financial transaction with OO programming (for different applications using relational database) because they all suggest a systematic method that use "instance" in a structural database to track all of the components of costs and fees each time a financial transaction is processed. It has been recognized that a finance system would be able to measure profitability in a flexible

manner and to measure the impact of any changes from banking clients by tracking those variables.

23. **Claim 70 is rejected under 35 U.S.C. §103(a) are rejected under 35 U.S.C. §103(a) as being unpatentable over Moore et al. (US Pat. 5,630,127), in view of Burt et al. (US Pat. 5,682,482).**

This claim further defines that "means for creating a second billing instance linked to said first production service instance by said 2<sup>nd</sup> service instance".

Moore et al. teach that an OOP software is used for created different instances that link with another instance (e.g. see Moore et al. Fig.4, and col.10 lines 25-55).

24. **Claims 49-52 are rejected under 35 U.S.C. §103(a) are rejected under 35 U.S.C. §103(a) as being unpatentable over Moore et al. (US Pat. 5,630,127), in view of Burt et al. (US Pat. 5,682,482).**

A. Re. To claim 49: This claim is directed to a method of pricing transactions containing similar limitations as in "system" claim 70. Therefore, similar rationales and references set forth are also used for a 35 USC 103(a) rejection.

B. Re. To claim 51: This claim is directed to a method of pricing transactions containing similar limitations as in "system" claim 72. Therefore, similar rationales and references set forth are also used for a 35 USC 103(a) rejection.

C. Re. To claim 52: This claim is directed to a method of pricing transactions containing similar limitations as in

"system" claim 73; therefore, similar rationales and references set forth are also used for a 35 USC 103(a) rejection.

D. Re. To claim 53: This claim is directed to a method of pricing transactions containing similar limitations as in "system" claim 74; therefore, similar rationales and references set forth are also used for a 35 USC 103(a) rejection.

E. Re. To claim 54: This claim is directed to a method of pricing transactions containing similar limitations as in "system" claim 75: therefore, similar rationales and references set forth are also used for a 35 USC 103(a) rejection.

25. **Claim 55 is rejected under 35 U.S.C. §103(a) are rejected under 35 U.S.C. §103(a) as being unpatentable over Moore et al. (US Pat. 5,630,127), in view of Burt et al. (US Pat. 5,682,482), further in view of Rothstein (US Pat. 5,636,117).**

The rationales and references for rejecting claim 54 are incorporated.

Rothstein further teaches that a market segment instance could be an entity instance (see 2:8-10, 2:54-47, 3:9-12) (e.g., mortgage entities are linked to business models by indices in a program).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to combine specific applications of Moore et al., and Burt et al., in OOP financial transaction with Rothstein because they all suggest a systematic

method that use "instance" in an OOP to track components of costs and fees each time a financial transaction is processed.

Artisan would recognize that an instant in OOP would be a variable to measure the impact of any changes from financial transactions by tracking those instance variables.

26. Claims 64, 66, and 85 are rejected under 35 U.S.C. §103(a) are rejected under 35 U.S.C. §103(a) as being unpatentable over Moore et al. (US Pat. 5,630,127), in view of Burt et al. (US Pat. 5,682,482), further in view of Claus et al. (US Pat. 5,559,313),

The rationales and references for rejecting claim 84 are incorporated.

Claus et al., further express analogous instances in a database, the examiner submits that since they are considered as variable instances in OOPs (see Figs. 6, 9-11, 13, 15) for analogous examples that were claimed about:

- an entity instance could be an account instance;
- an entity instance could be a client instance;
- an entity instance could be a market segment instance.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to combine specific applications to combine Moore et al., Burt et al., and Claus et al. in financial transaction with OO programming (for different applications using relational database) because they all suggest a systematic method that use "instance" in a structural database

to track all of the components of costs and fees each time a financial transaction is processed. It has been recognized that a finance system would be able to measure profitability in a flexible manner and to measure the impact of any changes from banking clients by tracking those variables.

27. Re. To claim 56: This claim is directed to a method of pricing transactions containing similar limitations as in "system" claim 76; therefore, similar rationales and references set forth are also used for a 35 USC 103(a) rejection.

28. Re. To claim 57: This claim is directed to a method of pricing transactions containing similar limitations as in "system" claim 77; therefore, similar rationales and references set forth are also used for a 35 USC 103(a) rejection.

29. Re. To claim 58: This claim is directed to a method of pricing transactions containing similar limitations as in "system" claim 78; therefore, similar rationales and references set forth are also used for a 35 USC 103(a) rejection.

30. Re. To claim 59: This claim is directed to a method of pricing transactions containing similar limitations as in "system" claim 79; therefore, similar rationales and references set forth are also used for a 35 USC 103(a) rejection.

31. Re. To claim 60: This claim is directed to a method of pricing transactions containing similar limitations as in

"system" claim 80; therefore, similar rationales and references set forth are also used for a 35 USC 103(a) rejection.

32. Re. To claim 61: This claim is directed to a method of pricing transactions containing similar limitations as in "system" claim 81; therefore, similar rationales and references set forth are also used for a 35 USC 103(a) rejection.

33. To claim 62: This claim is directed to a method of pricing transactions containing similar limitations as in "system" claim 82; therefore, similar rationales and references set forth are also used for a 35 USC 103(a) rejection.

34. Re. to claim 63: This claim is directed to a method of pricing transactions containing similar limitations as in "system" claim 83; therefore, similar rationales and references set forth are also used for a 35 USC 103(a) rejection.

35. Re. To claim 65: This claim is directed to a method of pricing transactions containing similar limitations as in "system" claim 84; therefore, similar rationales and references set forth are also used for a 35 USC 103(a) rejection.

36. Re. To claim 66: This claim is directed to a method of pricing transactions containing similar limitations as in "system" claim 85; therefore, similar rationales and references set forth are also used for a 35 USC 103(a) rejection.

37. Re. To claim 86: The rationales and references for rejecting claim 84 are incorporated.

- a limitation is: means for creating a second price table instance related to first entity instance.

Moore et al. teach that an OOP software is used for created different instances (i.e., a second price instance) that link with another instance (i.e., a first entity instance), (see Moore et al. Fig.4, and col.10 lines 25-55).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to combine specific applications to combine Moore et al., Burt et al., in financial transaction with OO programming (for different applications using relational database) because they all suggest a systematic method that use "instance" in a structural database to track all of the components of costs and fees each time a financial transaction is processed. It has been recognized that a finance system would be able to measure profitability in a flexible manner and to measure the impact of any changes from banking clients by tracking those variables.

38. Re. To claim 67: This claim is directed to a method of pricing transactions containing similar limitations as in "system" claim 86; therefore, similar rationales and references set forth are also used for a 35 USC 103(a) rejection.

**Conclusion**

39. Claims 47-86 are not patentable. The submitted amendment (received on 10/18/2004) necessitates new grounds for rejection. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicants are reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

**40. Remarks: Pending claims 47, 48, and 69 ... are also unpatentable over Carter, III (Pub. No. US 2002/0026368 A1).**

These claims are directed to steps/means of:

- creating a transaction instance (see Carter III, Figs.7, 8 for a database containing a charge on "Base Cost" of a transaction);
- creating a production service instance (see Carter III, Figs.7, 8 for a database containing "Shipping Charges" service);

- creating a billing service instance (see Carter III, Figs.7, 8 for a database containing a charge on "Maintenance" service); and
- pricing said transaction for billing a customer based on above related instances (i.e., obtaining a final price taking into account related items in a pricing table for above instances; see Carter III, the abstract).

As to claim 48: Carter III suggests about creating a service instance linked to said transaction instance by said first relation instance. It would have been obvious to create a second service instance by repetition (see Carter III, Figs.7, and 8).

41. These references are also considered having similar subject matters to this application:

- Durand et al., (US Pat. 5,694,598) teach that an account instance is represented as a data list similarly as an entity instance; a market segment instant could be an entity instance; for a use of instance in OOP in a relational database, wherein different programming instances can be linked together); their patent discloses: "The relational database model was introduced in the early 1970's by E. F. Codd. Since then, the relational model has become the model employed by most commercial database management systems (DBMS). Data in a relational database is represented as a collection of relations. Each relation can be

thought of as a table. Like the relational database model, *object-oriented* programming ("OOP") has also existed since the early 1970's. In the early 1990's, *object-oriented* programming gained widespread acceptance due to increased power of workstations, proliferation of graphical user-interfaces and the development of hybrid *object-oriented* languages such as C++.

The OOP paradigm provides a class construct which combines data and procedural abstractions. The definition of a class includes a definition of the storage requirements of the class as well as the procedures which define how objects of the class behave. An object is an instance of a class. Every object includes the data and procedural characteristics of its class. In addition, new objects inherit the storage and functionality defined by all classes used to define the parent of the object. The present proliferation of relational DBMSs coupled with the increasing popularity of the OOP paradigm has resulted in a desire to map data between data models. In particular, it is desirable to access relational databases in OOP applications, and to access *object-oriented* data from within a relational DBMS.

Commercial tools currently available for mapping *object-oriented* data to relational DBMSs include Persistence, ROCK Phase II, and ObjectStore. These tools are primarily intended to allow application objects to be persistent. Further, these applications typically assume a straight mapping correspondence between application objects and a database schema. Various approaches have been considered for object-relational

integration. In most approaches, the purpose has been to interface *object-oriented* applications with relational data storage.

- Vic Arnold et al., US-PAT-NO: 5,936,860 - 8/10/1999 - 700/95, Object oriented technology framework for warehouse control, (see 14:31-40, 15:4-22), wherein the patent teaches that a state is encoded in instance variables (data members in an OOP); banking transactions and related pricing were known to implement this model for the pending claimed relational structures (such use of a relational database management software have been applied for banking transactions, e.g., **these para. indicate relationships of an object and access type with the value in the object instance entity**); (see the abstract, 13:56 to 14:5); and an **instance variable/data member is data which an object keeps track of** (see 16:57-65).

42. Any inquiry concerning this communication or earlier communications from the examiner should be directed to CUONG H. NGUYEN whose telephone number is 703-305-4553. The examiner can normally be reached on 7:15 am - 3:45 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, THOMAS G. BLACK can be reached on 703-305-8233. The fax phone number for the organization where this application is assigned is 703-305-7687.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

*Cuongnguyen*

CUAN  
CUONG H. NGUYEN  
Primary Examiner  
Art Unit 3661